MICROCONTROLLER AND INTERFACING LAB
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING, BUET

Experiment : 9

LCD interface with 8086

1.1 Objective

The objective of this experiment are

- To know the communication protocol with LCD
- Establish the protocol between LCD and 8086
- Test different features of LCD

1.2 Learning Outcome

At the end of the experiment students will be able to

- Communicate with LCD
- Can display anything in the LCD

1.3 Liquid Crystal Display (LCD) Module

LCD is a thin, flat display device made up of pixels arrayed in front of a light source or reflector. Uses very small amounts of electric power, and is therefore suitable for use in battery-powered electronic devices.

1.3.1 Characteristics

Each module contains a CMOS controller and all necessary drivers which have low power consumption. The controller is equipped with an internal character generator ROM, RAM and RAM for display data. All display functions are controllable by instructions making interfacing practical.

Register:

The Controller has two 8 bit registers, the Instruction register (IR) and the data register (DR).
The IR is a write only register to store instruction codes like Display Clear or Cursor Shift as well as addresses for the Display Data RAM (DD RAM) or the Character Generator RAM (CG RAM).
The DR is a read/write register used for temporarily storing data to be read/written to/from the DD RAM or CG RAM. Data written into the DR is automatically written into DD RAM or CG RAM by an internal operation of the display controller. The DR is also used to store data when reading out data from DD RAM or CG RAM. When address information is written into IR, data is read out from DD RAM or CG RAM to DR by an internal operation. Data transfer is then completed by reading the DR. After performing a read from the DR, data in the DD RAM or CG RAM at the next address is sent to the DR for the next read cycle. The register select (RS) signal determines which of these two registers is selected.

Pin Number	Symbol
1	Vss
2	Vcc
3	Vee
4	RS
5	R/W
6	E
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7

RS	R/W	Operation
0	0	IR write, internal operation (Display Clear etc.)
0	1	Busy flag (DB7) and Address Counter (DB0 ~ DB6) read
1	0	DR Write, Internal Operation (DR ~ DD RAM or CG RAM)
1	1	DR Read, Internal Operation (DD RAM or CG RAM)

Busy Flag :

When the busy flag is high or “1” the module is performing an internal operation and the next instruction will not be accepted. As shown in Table 1.4, the busy flag outputs to DB7 when RS=0 and a read operation is performed. The next instruction must not be written until ensuring that the busy flag is low or “0”.

Address Counter (AC)

The address counter (AC) assigns addresses to the DD RAM and the CG RAM. When the address of an instruction is written into the IR, the address information is sent from the IR to the AC. The selection of either DD RAM or CG RAM is also determined concurrently by the same instruction. After writing into or reading from the DD RAM or CG RAM the address counter (AC) is automatically incremented by 1 or decremented by 1 (determined by the I/D bit in the “Entry Mode Set” command.) AC contents are output to DB0 ~ DB7 when RS = 0 and a read operation is performed, as shown in Table.

Display Data RAM (DD RAM)

The Display Data RAM (DD RAM) stores the display data represented in 8 bit character codes. Its capacity is 80 x 8 bits or 80 characters. The Display Data RAM that is not used for the display an be used as a general data RAM.

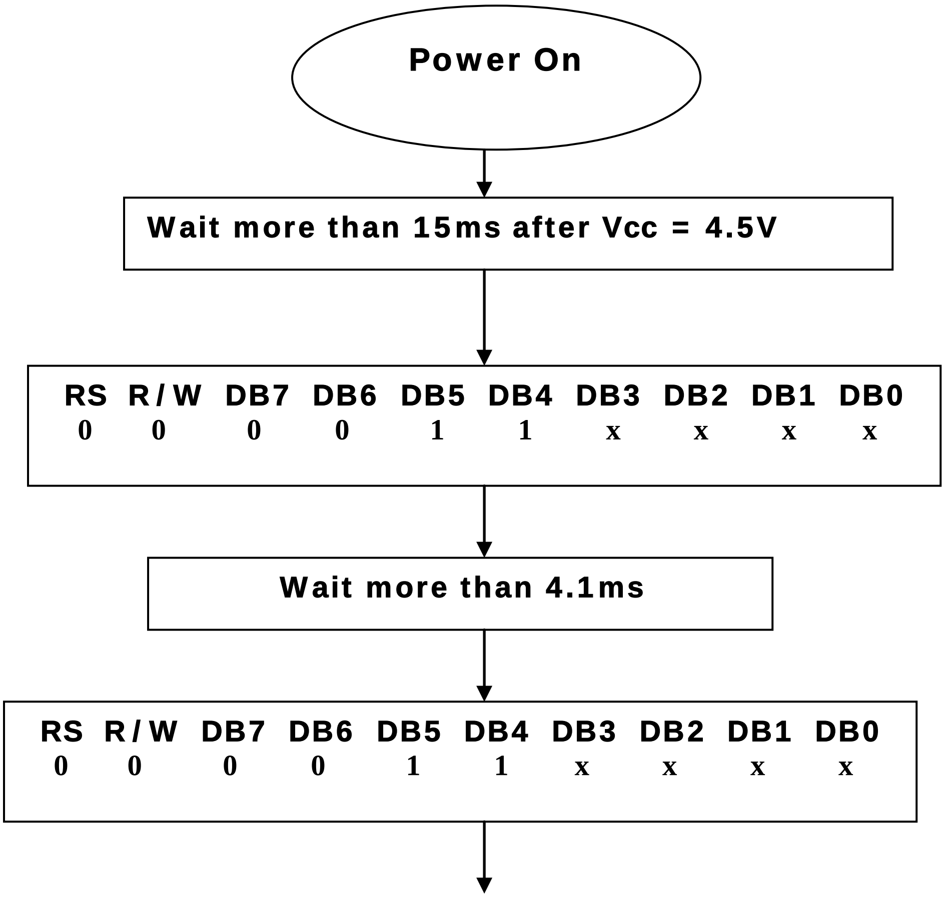
MPU INTERFACING

Each character display can be operated in either 4 or 8 bit mode. Instructions/Data are written to the display using the signal timing characteristics found below.

When operating in 4 bit mode, data is transferred in two 4 bit operations using data bits DB4 - DB7. DB0 - DB3 are not used and should be tied low. When using 4 bit mode, data is transferred twice before the instruction cycle is complete. First the high order nibble is transferred then the low order nibble. The busy flag should only be checked after both nibbles are transferred. When operating in 8 bit mode, data is transferred using the full 8 bit bus DB0 - DB7.

Software Initialization

8 - Bit Initialization:



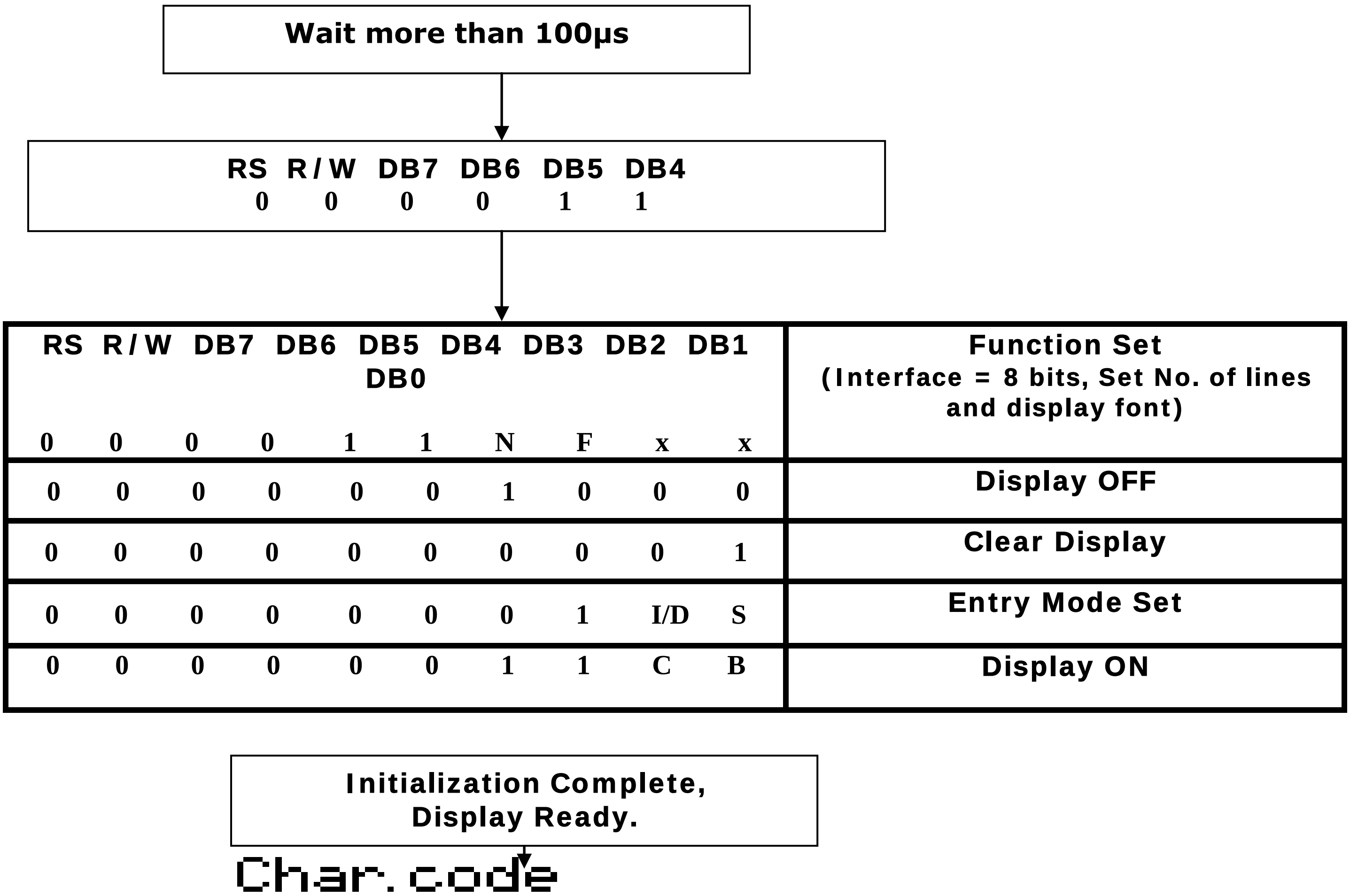
Power On

Wait more than 15ms after Vcc = 4.5V

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	x	x	x	x

Wait more than 4.1ms

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	x	x	x	x



Wait more than 100μs

RS	R/W	DB7	DB6	DB5	DB4
0	0	0	0	1	1

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	0	0	0	1	1	N	F	x	x	Function Set (Interface = 8 bits, Set No. of lines and display font)
0	0	0	0	0	0	1	0	0	0	Display OFF
0	0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	1	I/D	S	Entry Mode Set
0	0	0	0	0	0	1	1	C	B	Display ON

Initialization Complete, Display Ready.

Char.code

	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000			0	@	P	`	p		ー	タ	ミ	α	p
xxxx0001		!	1	A	Q	a	q	｡	ア	チ	ム	ä	q
xxxx0010		"	2	B	R	b	r	｢	イ	ツ	メ	β	θ
xxxx0011		#	3	C	S	c	s	｣	ウ	テ	モ	ε	∞
xxxx0100		$	4	D	T	d	t	､	エ	ト	ヤ	μ	Ω
xxxx0101		%	5	E	U	e	u	･	オ	ナ	ユ	σ	ü
xxxx0110		&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111		'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
xxxx1000		(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
xxxx1001		)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010		*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
xxxx1011		+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
xxxx1100		,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
xxxx1101		-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
xxxx1110		.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
xxxx1111		/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

instruction set

Instruction	Code										Description	Execution time**
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears display and returns cursor to the home position (address 0).	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position (address 0). Also returns display being shifted to the original position. DDRAM contents remains unchanged.	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.	40uS
Display On/Off control	0	0	0	0	0	0	1	D	C	B	Sets On/Off of all display (D), cursor On/Off (C) and blink of cursor position character (B).	40uS
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	*	*	Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged.	40uS
Function set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length	40uS

instruction set												
Instruction	Code										Description	Execution time**
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
											(DL), number of display line (N) and character font(F).	
Set CGRAM address	0	0	0	1	CGRAM address						Sets the CGRAM address. CGRAM data is sent and received after this setting.	40uS
Set DDRAM address	0	0	1	DDRAM address							Sets the DDRAM address. DDRAM data is sent and received after this setting.	40uS
Read busy-flag and address counter	0	1	BF	CGRAM / DDRAM address							Reads Busy-flag (BF) indicating internal operation is being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction).	0uS
Write to CGRAM or DDRAM	1	0	write data								Writes data to CGRAM or DDRAM.	40uS
Read from CGRAM or DDRAM	1	1	read data								Reads data from CGRAM or DDRAM.	40uS

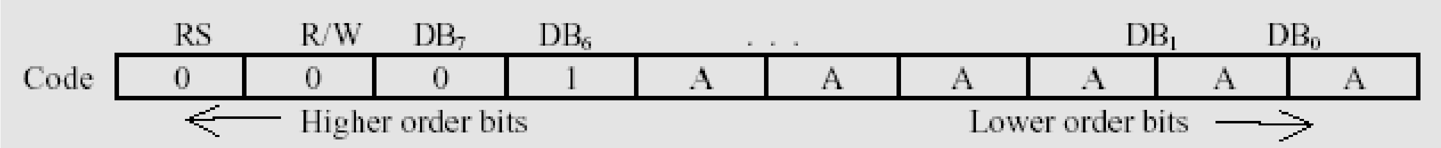
Remarks:

- DDRAM = Display Data RAM.
- CGRAM = Character Generator RAM.
- DDRAM address corresponds to cursor position.
- * = Don't care.
- ** = Based on Fosc = 250KHz.

Table: Bit names		
Bit name	**Settings**	
I/D	0 = Decrement cursor position	1 = Increment cursor position
S	0 = No display shift	1 = Display shift
D	0 = Display off	1 = Display on
C	0 = Cursor off	1 = Cursor on
B	0 = Cursor blink off	1 = Cursor blink on
S/C	0 = Move cursor	1 = Shift display
R/L	0 = Shift left	1 = Shift right
DL	0 = 4-bit interface	1 = 8-bit interface
N	0 = 1/8 or 1/11 Duty (1 line)	1 = 1/16 Duty (2 lines)
F	0 = 5x7 dots	1 = 5x10 dots
BF	0 = Can accept instruction	1 = Internal operation in progress

Detailed Explanations

Set CG RAM Address

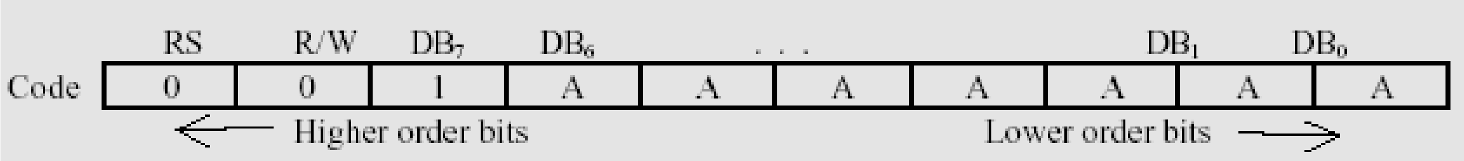


	RS	R/W	DB_7	DB_6		. . .			DB_1	DB_0
Code	0	0	0	1	A	A	A	A	A	A

← Higher order bits Lower order bits →

Sets the address counter to the CG RAM address AAAAAAA. Data is then written/read to from the CG RAM.

Set DD RAM Address



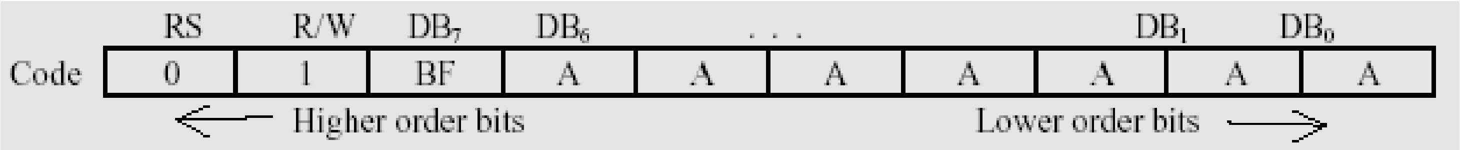
	RS	R/W	DB_7	DB_6		. . .			DB_1	DB_0
Code	0	0	1	A	A	A	A	A	A	A

← Higher order bits Lower order bits →

Sets the address counter to the DD RAM address AAAAAAA. Data is then written/read to from the DD RAM.

For a 1-line display module AAAAAAA is “00” ~ “4F” (hexadecimal). For 2-line display module AAAAAAA is “00” ~ “27” (hexadecimal) for the first line and “40” ~ “67” (hexa decimal) for the second line.

Read Busy Flag and Address



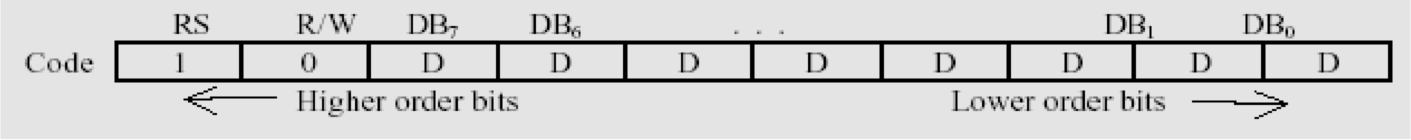
	RS	R/W	DB_7	DB_6		. . .			DB_1	DB_0
Code	0	1	BF	A	A	A	A	A	A	A

← Higher order bits Lower order bits →

Reads the busy flag (BF) and value of the address counter (AC). BF = 1 indicates that on internal operation is in progress and the next instruction will not be accepted until BF is set to “0”. If the display is written while BF = 1, abnormal operation will occur.

The BF status should be checked before each write operation.
At the same time the value of the address counter expressed in binary AAAAAAA is read out. The address counter is used by both CG and DD RAM and its value is determined by the previous instruction.

Write Data to CG or DD RAM



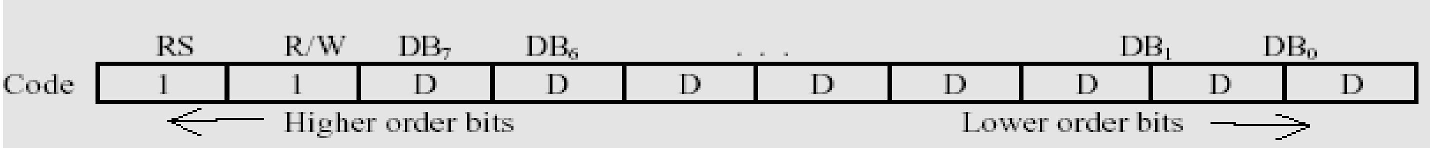
	RS	R/W	DB_7	DB_6		. . .			DB_1	DB_0
Code	1	0	D	D	D	D	D	D	D	D

← Higher order bits Lower order bits →

Writes binary 8-bit data DDDDDDDD to the CG or DD RAM.

The previous designation determines whether the CG or DD RAM is to be written (CG RAM address set or DD RAM address set). After a write the entry mode will automatically increase or decrease the address by 1. Display shift will also follow the entry mode.

Read Data from CG or DD RAM



	RS	R/W	DB_7	DB_6		. . .			DB_1	DB_0
Code	1	1	D	D	D	D	D	D	D	D

← Higher order bits Lower order bits →

Reads binary 8-bit data DDDDDDDD from the CG RAM or DD RAM.

The previous designation determines whether the CG or DD RAM is to be read.

Before entering the read instruction, you must execute either the CG RAM or DD RAM address set instruction.
If you don't, the first read data will be invalidated. When serially executing the "read" instruction the next address data is normally read from the second read.

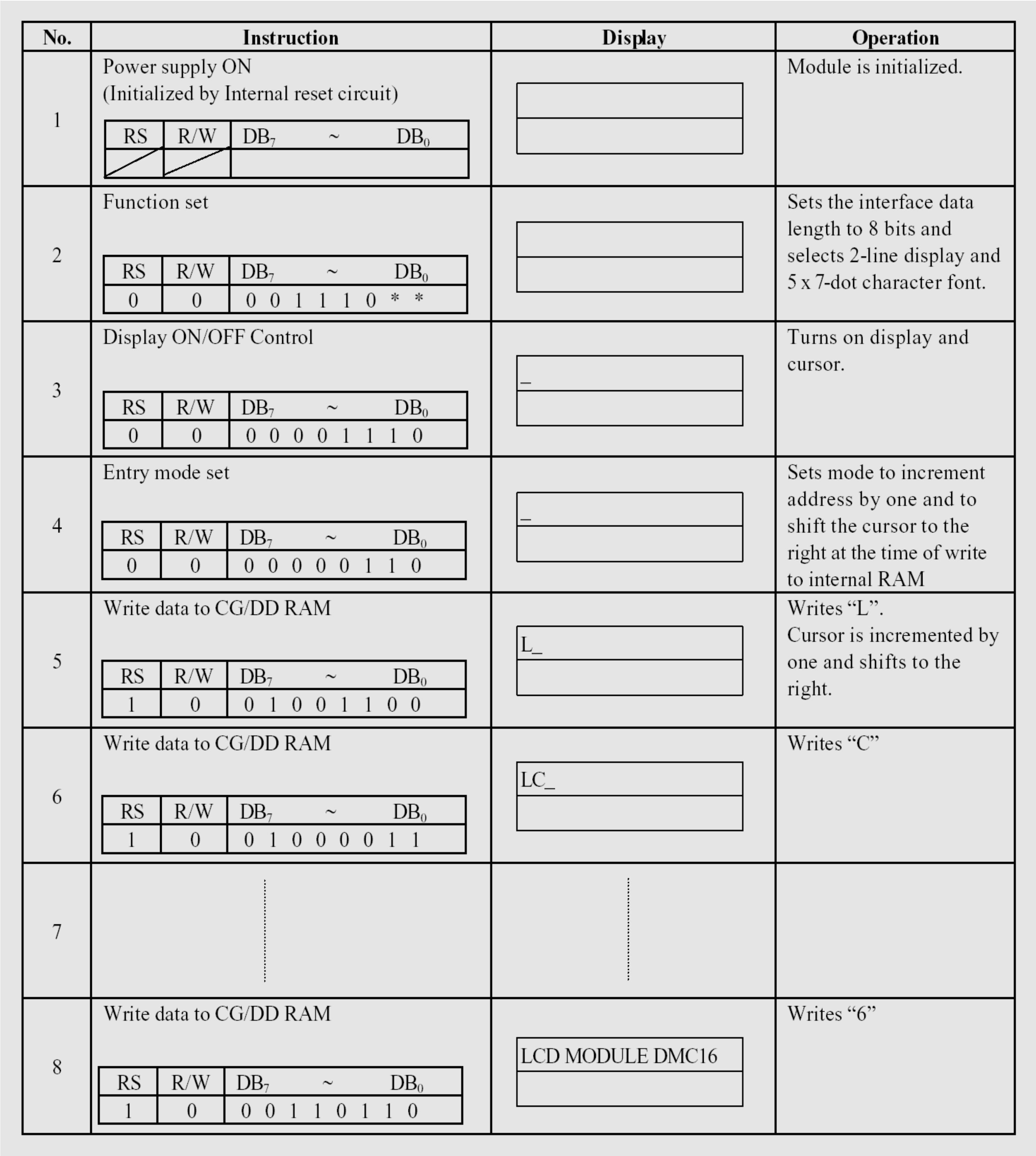
The "address set" instruction need not be executed just before the "read" instruction when shifting the cursor using cursor instruction (when reading DD RAM). The cursor shift instruction operation is the same as that of the DD RAM address set instruction.

After a read, the entry mode automatically increases or decreases the address by 1; however, display shift is not executed no matter what the entry mode is.

Note: The address counter (AC) is automatically incremented or decremented by 1 after a "write" instruction to either CG RAM or DD RAM. RAM data selected by the AC cannot then be read out even if "read" instructions are executed.

The conditions for correct data reads are: (a) Execute either the address set instruction or cursor shift instruction (only with DD RAM) or (b) The execution of the "read data" instruction from the second time when the read instruction is performed multiple times in serial.

16 Character x 2 Line Display



No.	Instruction	Display	Operation
1	Power supply ON (Initialized by Internal reset circuit) RS R/W DB$_7$ ~ DB$_0$		Module is initialized.
2	Function set RS R/W DB$_7$ ~ DB$_0$ 0 0 0 0 1 1 1 0 * *		Sets the interface data length to 8 bits and selects 2-line display and 5 x 7-dot character font.
3	Display ON/OFF Control RS R/W DB$_7$ ~ DB$_0$ 0 0 0 0 0 0 1 1 1 0	_	Turns on display and cursor.
4	Entry mode set RS R/W DB$_7$ ~ DB$_0$ 0 0 0 0 0 0 0 1 1 0	_	Sets mode to increment address by one and to shift the cursor to the right at the time of write to internal RAM
5	Write data to CG/DD RAM RS R/W DB$_7$ ~ DB$_0$ 1 0 0 1 0 0 1 1 0 0	L_	Writes “L”. Cursor is incremented by one and shifts to the right.
6	Write data to CG/DD RAM RS R/W DB$_7$ ~ DB$_0$ 1 0 0 1 0 0 0 0 1 1	LC_	Writes “C”
7	⋮	⋮	
8	Write data to CG/DD RAM RS R/W DB$_7$ ~ DB$_0$ 1 0 0 0 1 1 0 1 1 0	LCD MODULE DMC16	Writes “6”

No.	Instruction	Display	Operation
9	Set DD RAM address. RS R/W DB$_7$ ~ DB$_0$ 0 0 1 1 0 0 0 0 0 0	LCD MODULE DMC16 _	Sets RAM address so that the cursor is positioned at the head of the 2nd line.
10	Write data to CG/DD RAM RS R/W DB$_7$ ~ DB$_0$ 1 0 0 0 1 1 0 0 1 1	LCD MODULE DMC16 1_	Write “1”
11	Write data to CG/DD RAM RS R/W DB$_7$ ~ DB$_0$ 1 0 0 0 1 1 0 0 1 0	LCD MODULE DMC16 16_	Writes “6”
12	⋮	⋮	
13	Write data to CG/DD RAM address RS R/W DB$_7$ ~ DB$_0$ 1 0 0 0 1 0 1 0 1 0	LCD MODULE DMC16 16 DIGITS, 2 LINES_	Writes “S”
14	Set DD/RAM address RS R/W DB$_7$ ~ DB$_0$ 0 0 1 0 0 0 0 0 0 0	LCD MODULE DMC16 16 DIGITS, 2 LINES	Moves cursor to original position
15	Clear display RS R/W DB$_7$ ~ DB$_0$ 0 0 0 0 0 0 0 0 0 1	_	Return both display and cursor to the original position
16	⋮	⋮	

Exercise 1

```
;****************************************
;   *     LCD Interface EXPERIMENT PROGRAM     *
;   *              8086 Training kit              *
;   *     EEE Department, BUET                    *
;****************************************
;
;Code for MDA 8086 . NOTE : activate 4 lines below for MDA 8086 if not activated
;       STACK       EQU   0A00H       ; STACK POINTER ; A 0 Must be added before A00H
;       IR_WR       EQU   00H         ; USED TO WRITE INSTRUCTION REGISTER
;       ST_RD       EQU   02H         ; USED TO READ  STATUS
;       DR_WR       EQU   04H         ; USED TO WRITE DISPLAY DATA RAM
;
;Code for MTS-86C (Black processor) . NOTE : activate 4 lines below for MTS-86C(Black) if not
;activated
;       IR_WR       EQU   3FF9H       ; USED TO WRITE INSTRUCTION REGISTER
;       ST_RD       EQU   3FFBH       ; USED TO READ  STATUS
;       DR_WR       EQU   3FFDH       ; USED TO WRITE DISPLAY DATA RAM
;       STACK       EQU   0A00H       ; STACK POINTER ; A 0 Must be added before A00H
;                         ; because address value starting with non numerical
;                         ; value creates error while compiling
;
;Code for MTS-86C (White Processor) . NOTE : activate 4 lines below for MTS-86C(White) if not
;activated
;       IR_WR       EQU   0FFC1H      ; USED TO WRITE INSTRUCTION REGISTER
;       ST_RD       EQU   0FFC3H      ; USED TO READ  STATUS
;       DR_WR       EQU   0FFC5H      ; USED TO WRITE DISPLAY DATA RAM
;       STACK       EQU   0A00H       ; STACK POINTER ; A 0 Must be added before A00H
;                         ; because address value starting with non numerical
;                         ; value creates error while compiling
;

;code common for both MDA and MTS starts from here
CODE  SEGMENT
      ASSUME      CS:CODE,DS:CODE,ES:CODE,SS:CODE
      ;
      ;
START:      ORG   0H    ; Use 1000H for MDA series
            MOV   AX,CS
            MOV   DS,AX ; Making the DS (Data Segment) and CS (Code Segment)
                              ;value same
            ;
            MOV   SS,AX ; Making the SS (Stack Segment) also same with CS, DS
            MOV   SP,STACK
            ;
            CALL  ALLCLR
            ;
            CALL  LN11
            MOV   SI,OFFSET LINE1
            CALL  STRING
            ;
            CALL  LN21
            MOV   SI,OFFSET LINE2
            CALL  STRING
            ; Blinks the whole display
BLINK:CALL  DISPOFF
            CALL  TIMER
            CALL  DISPON
            CALL  TIMER
            JMP   BLINK
      ;
```

```
LINE1  DB      'Hi BUET Students !',00H,00H
LINE2          DB      'Make me friend.',00H,00H
        ;
        ; LCD instruction
ALLCLR:         MOV     AH,00000001B ; Clears entire display
                JMP     OUT1
        ;
DISPOFF:
                MOV     AH,00001000B ; Display off, cursor off, not blink
                JMP     OUT1
        ;
DISPON:         MOV     AH,00001111B ; Display on, cursor on, cursor blink
                JMP     OUT1
        ;
LN11:   MOV     AH,00000010B ; Returns to home position
                JMP     OUT1
        ;
LN21:   MOV     AH,11000000B ; Sets RAM address so that the cursor is positioned
                            ; at the head of the 2nd line.
                JMP     OUT1
        ;
        ;To write to instruction register
OUT1:   PUSH            AX
                PUSH            DX
                CALL            BUSY
                MOV             AL,AH
                MOV             DX,IR_WR
                OUT             DX,AL
                POP             DX
                POP             AX
                RET
        ; busy flag check, must be done before any write operation
BUSY:   PUSH            DX
                PUSH            AX
                MOV             DX,ST_RD
BUSY1:          IN              AL,DX
                AND             AL,10000000B
                JNZ             BUSY1
                POP             AX
                POP             DX
                RET
                ;
        ;To send a single character
CHAROUT:
                PUSH    DX
                PUSH            AX
                CALL            BUSY
                MOV             AL,AH
                MOV             DX,DR_WR
                OUT             DX,AL
                POP             AX
                POP             DX
                RET
        ;To out a string line from address CS:[SI]
STRING:         MOV             AH,BYTE PTR CS:[SI]
                CMP             AH,00H
                JE              STRING1
                ;
                CALL            BUSY
                CALL            CHAROUT
                INC             SI
                JMP             STRING
STRING1:
                RET
                ;Timer Makes delay
TIMER:          PUSH            CX
                MOV             CX,0FFFFH
TIMER1:         DEC             CX
                JNZ             TIMER1
                POP             CX
                RET
                ;
CODE            ENDS
                END     START
```

Exercise 2

Using the code above make a program that will

a) display your group no. in the first line and roll no in the second line.
b) after some delay it will shift the line to right 3 space
c) after some delay it will shift the line to left 5 space
d) after some delay it will clear the whole display and cursor to the home position
e) after some delay it will repeat all of above actions

Exercise 3

Write a program that will count from 0 to 1000 and after counting to 1000 it will repeat it again. Display the results in the LCD.

Home Task:

1. Complete all the code for exercise 2,3 and explain all the steps done in the program.
2. Write a program that will display your group members name in the first line and roll in the second line and after some delay it will show next member's name and roll and repeatedly do it after finishing all member's task.

Prepared by : RAJIB MIKAIL , Lecturer, EEE, BUET . **Date: June' 07**